

PRIMA PARTE

1. **[5 punti]** Si consideri una coda con priorità P , implementata tramite *heap*, inizialmente vuota. Si supponga di eseguire le seguenti operazioni nell'ordine dato: $P.insert(5, \cdot)$, $P.insert(3, \cdot)$, $P.insert(6, \cdot)$, $P.insert(4, \cdot)$, $P.insert(7, \cdot)$, $P.insert(2, \cdot)$, $P.removeMin()$. Disegnare lo heap risultante (come albero, riportando solo le chiavi) dopo tutte le operazioni
2. **[6 punti]** Dimostrare che un Multi-Way Search Tree contenente n entry ha $n + 1$ foglie.
3. **[5 punti]** Il seguente algoritmo ordina una sequenza S di n interi.

```
S1 ← interi pari di S; S2 ← interi dispari di S;  
Ordina S1 con MergeSort;  
Ordina S2 con QuickSort deterministico;  
Merge(S1, S2; S);
```

Osservando che la separazione di pari e dispari può essere fatta in tempo $\Theta(n)$, determinare la complessità al caso pessimo dell'algoritmo esprimendola con $\Theta(\cdot)$.

SECONDA PARTE

1. **[9 punti]** Sia T un albero binario di ricerca le cui entry rappresentano studenti di un'università. Ogni studente è associato a una entry (k, x) , dove k è la matricola, e x indica se lo studente è straniero ($x = 1$) o italiano ($x = 0$). Per ogni nodo $v \in T$ esiste un intero $v.\text{numStr}$ che riporta il numero di studenti stranieri in T_v (sottoalbero con radice v). Progettare un algoritmo *ricorsivo* $\text{MinMatStraniero}(T, v)$ che dato un nodo $v \in T$ restituisce *la più piccola matricola di uno studente straniero in T_v* , e analizzarne la complessità. Se non ci sono studenti stranieri in T_v l'algoritmo restituisce `null`.
2. **[7 punti]** Sia $G = (V, E)$ un grafo *non connesso* con n vertici ed m archi. Progettare un algoritmo che conti le coppie di vertici $u, v \in V$ tali che u e v sono raggiungibili uno dall'altro tramite cammini, analizzandone la complessità. Per avere punteggio pieno la complessità deve essere $O(n + m)$. (Si ricordi che da un insieme di K oggetti si possono formare $K(K - 1)/2$ coppie distinte.)

TEMPO COMPLESSIVO A DISPOSIZIONE: 1.5 ore